Dossier de Mise en Œuvre



Réalisé par:

Baptiste Cahuziere - Cédric Cognard - Clément Peytureau - Dorian Caup - Romain Gaillard

Version numérique :
http://urlz.fr/pIw





Table des matières

I. Installation du serveur Jboss – Dorian Caup :

1) Pour bien commencer :

a. Téléchargement du serveur Jboss :

Rendez-vous sur : http://www.jboss.org/jbossas/downloads/ et téléchargez la dernière version du serveur Jboss.
Evitez de télécharger les versions X.0.

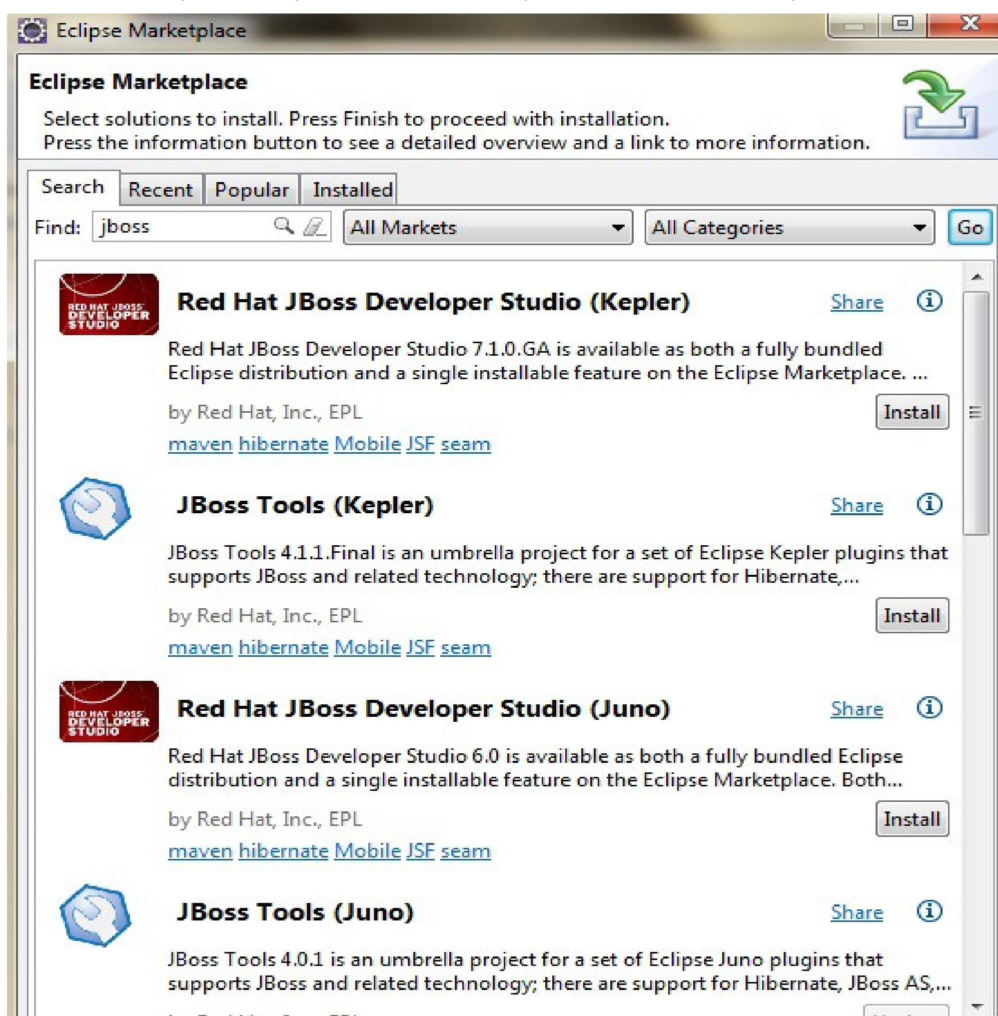
b. Installation du serveur Jboss :

Choisissez un répertoire définitif pour l'installation de votre serveur Jboss. Notez le path de l'installation du serveur.

2) Mise en oeuvre sur Eclipse :

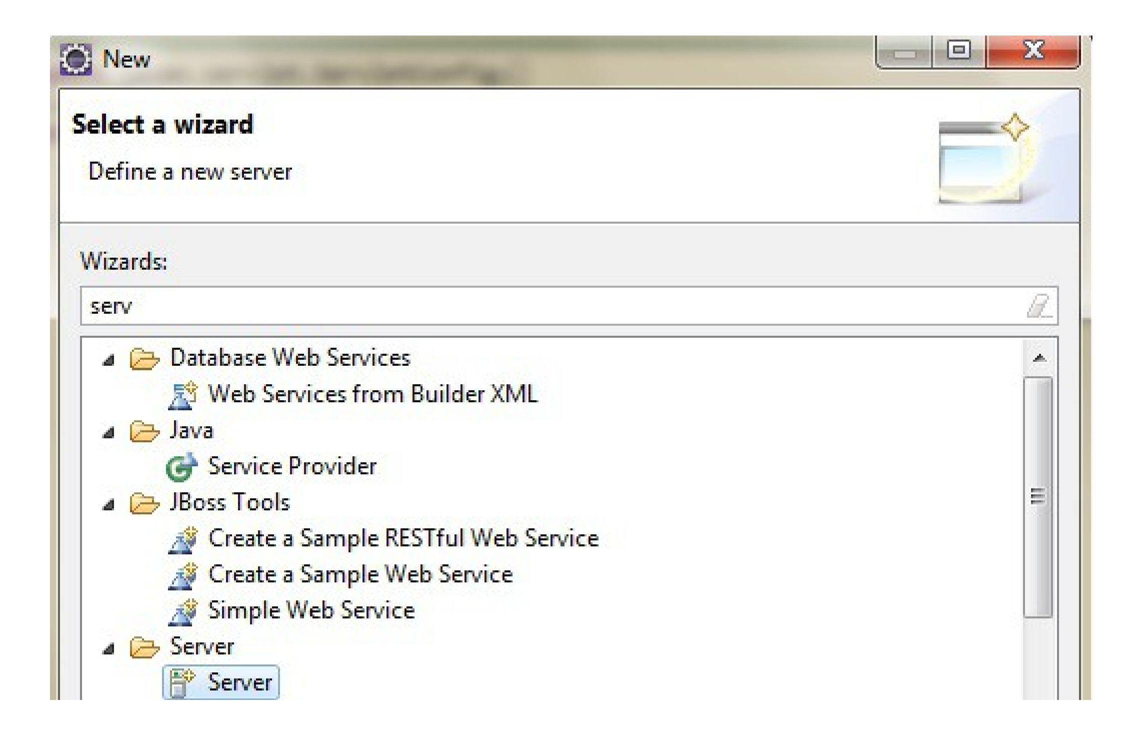
a. Téléchargement de Jboss Tools :

Rendez-vous sur Eclipse.
Pensez à noter voter version d'Eclipse (Juno,Kepler...).
Aller dans 'Help' -> 'Eclipse MarketPlace' puis recherchez : Cliquez sur 'Install' :



2)

Création du serveur :



Assignez le serveur au path du serveur Jboss précédemment installé.
Créer un nouveau projet 'Dynamic Web Project'.

Vous êtes prêt à lancer vôtre serveur.

3) Lancement du serveur :

Le serveur n'est pas configuré de base pour écouter toutes les entrées.

a. Lancer le serveur en local :

Dans le dossier bin se trouve tous les scripts de lancement du serveur. L'exécutable "standalone.bat" (standalone.sh sous linux/unix) permet de lancer une instance du serveur JBoss.

Lancez le script et rendez-vous sur la page d'accueil du serveur sur le port 8080 :
http://localhost:8080/

b. Ouvrir toutes les entrées :

Modifiez directement les fichiers de configuration.
Dans le dossier 'path\standalone\configuration\standalone.xml.' Modifiez la ligne :

```
<inet-address value="${jboss.bind.address:127.0.0.1}" />
```

Par :

```
<inet-address value="${jboss.bind.address:0.0.0.0}" />
```

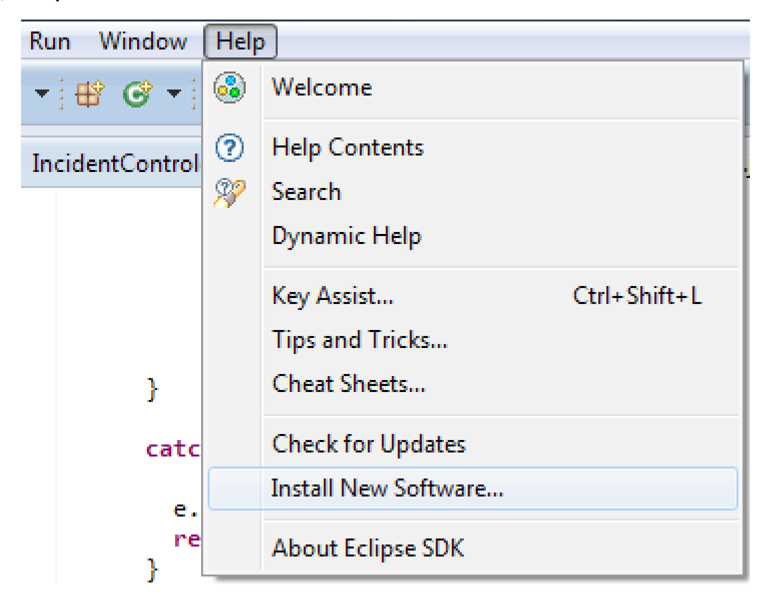
II. Installation d'Eclipse pour l'interface gestionnaire – Clément Peytureau:

Pour réaliser l'interface gestionnaire, nous avons eu besoin de deux logiciels différents : Eclipse v4.2 équipé de son plug-in e(fx)clipse v0.8.1 et de SceneBuilder v1.1. Il nous faut également Java JDK 7 comprenant JavaFX v2.2, ainsi que l'intégration de la librairie Axis v1.4 qui est sa version client.

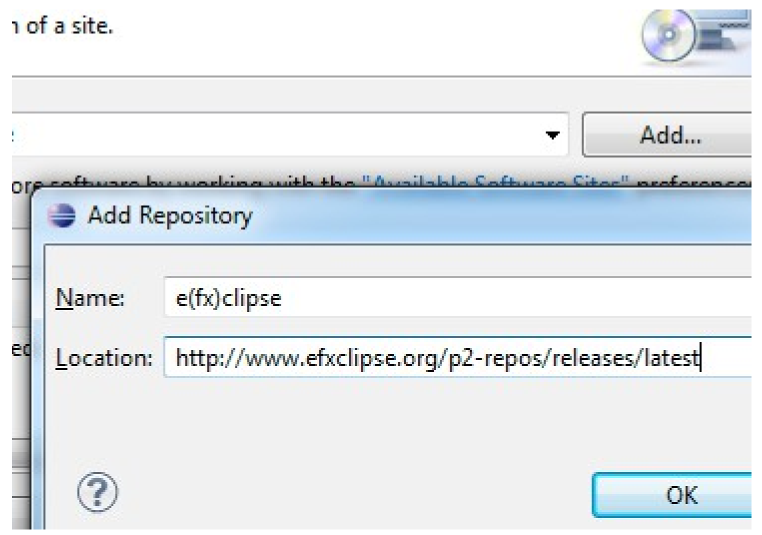
1) Installation d'Eclipse 4.2 + e(fx)clipse 0.8.1 :

Pour installer le plug-in e(fx)clipse, il faut télécharger Éclipse v4.2 sur **download.eclipse.org**, une fois Eclipse installé et lancé :

Dans l'onglet "Help", cliquer sur "Install New Software" :



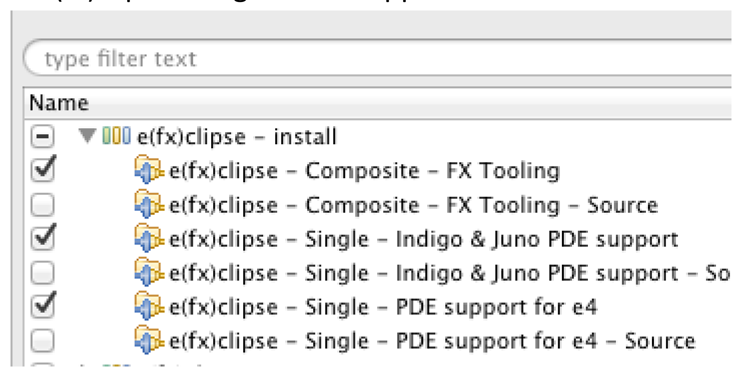
Eclipse ne connais pas la localisation du répertoire e(fx)clipse, il nous faut donc ajouter un lien vers le site d'e(fx)clipse :



http://downloads.efxclipse.org/p2-repos/releases/latest/

Il faut maintenant installer les options en cochant :

- e(fx)clipse - Composite - FX Tooling.
- e(fx)clipse - Single - Indigo & Juno PDE Support.
- e(fx)clipse - Single - PDE support for e4

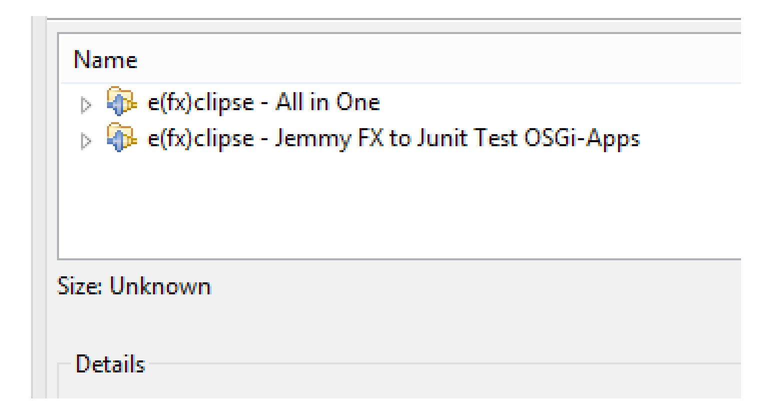


S'assurer que "Contact all update sites during install to find required software" soit bien coché.

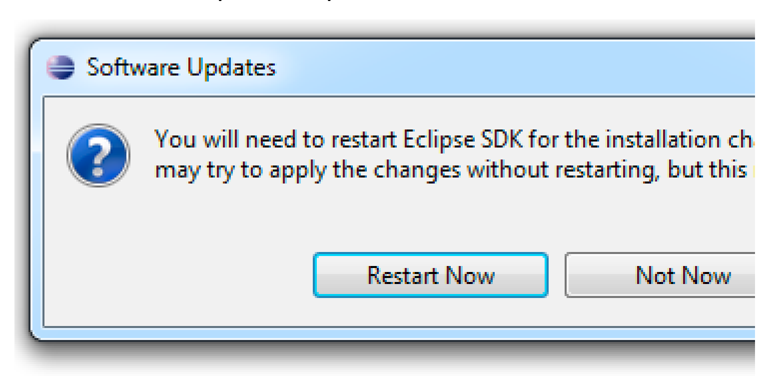
Il n'est pas recommandé d'installer les options suivantes qui sont censés être déjà présente dans Eclipse :

- e(fx)clipse - runtime - FX Runtime.
- e(fx)clipse - tooling - Any single component standalone.

Lancer l'installer et le laisser travailler :



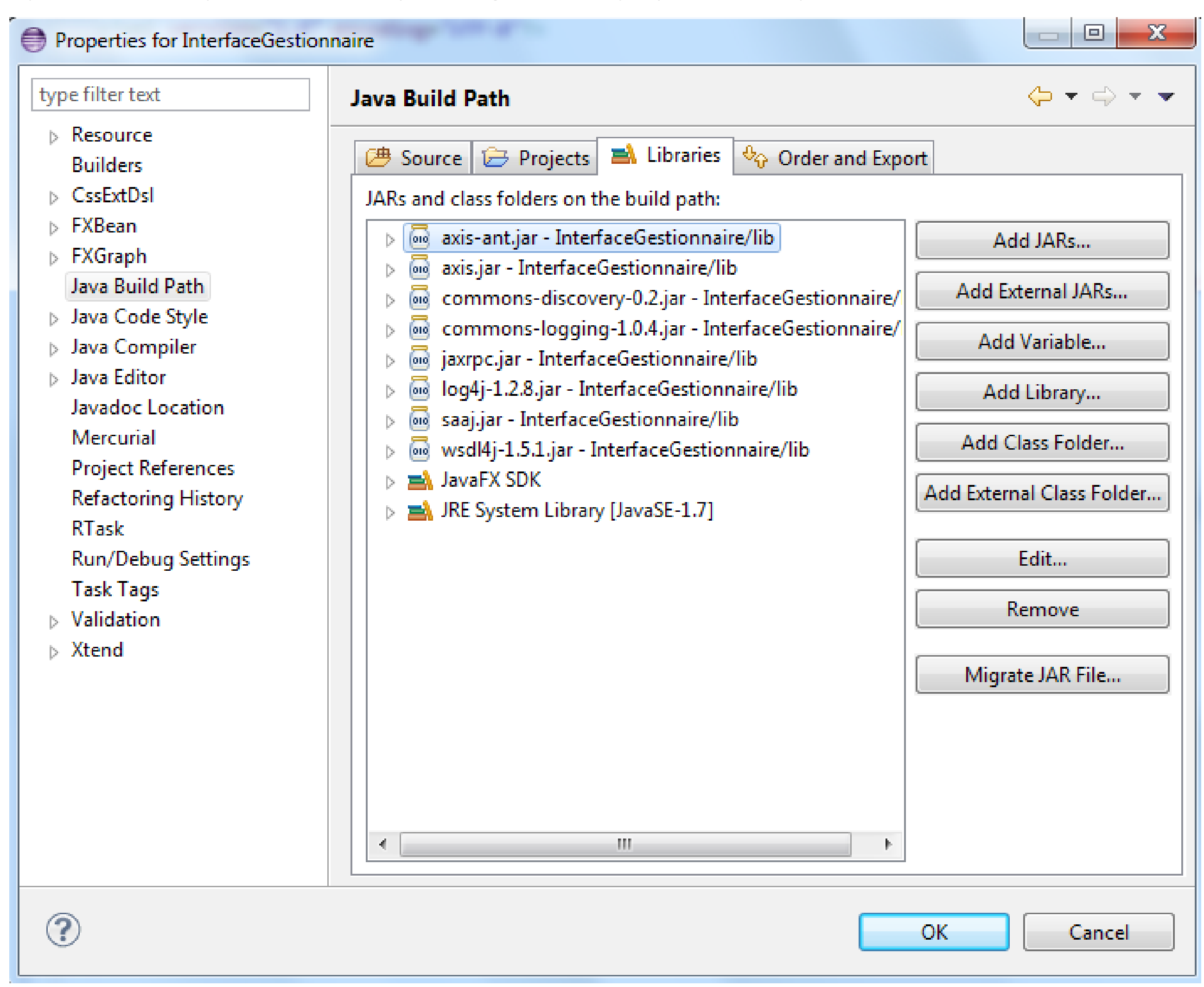
Redémarrer Eclipse lorsque demandé :



2) Implémentation librairie Axis 1.4 :

Pour pouvoir intégrer cette librairie il faut d'abord la télécharger sur le site http://axis.apache.org/axis/.

Après avoir récupéré la librairie, je l'intègre à mon projet sous Éclipse.



La librairie Axis 1.4 intègre le protocole SOAP (Simple Object Access Protocol) qui va nous permettre de communiquer et d'échanger des données avec le web service.

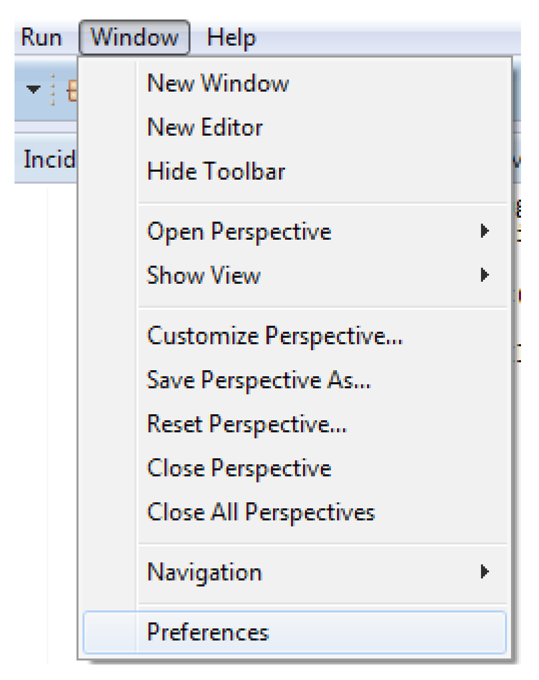
3) Installation de SceneBuilder 1.1 :

Les interfaces utilisées pour le language JavaFX sont construites par le biais du SceneBuilder par un système de drag&drop, il fait ensuite lui-même la transition vers un fichier FXML dont nous aurons ensuite l'utilité pour la création des actions sur notre interface.

Télécharger SceneBuilder et cliquer sur "**Scene Builder 1.1 Developer Preview**" et à l'installer.

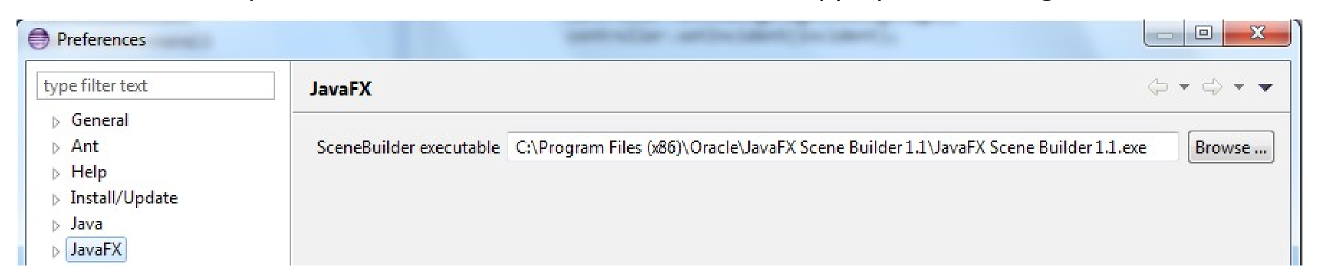
Une fois installée nous aurons besoin de l'implémenter dans notre environnement de

développement :



Dans l'onglet "Window", cliquer sur "Preferences" :

Entrer le chemin qui mène à l'exécutable du SceneBuilder et appliquer les changements.



Voilà, il ne reste plus qu'à créer un nouveau projet JavaFX.

4) Création d'un projet JavaFX :

Tout d'abord il nous faut créer un nouveau projet :

Pour se faire aller dans *File | New | Other | JavaFX project*, tout en spécifiant le nom du nouveau projet.

Il nous faut ensuite créer deux packages pour séparer les fichiers XML (affichage), des classes java classiques (code brut) :

Pour créer un package, clique droit sur le dossier *src | New | Package*.

5) Créer son premier fichier XML :

Il y a deux façons de créer une interface utilisateur. Utiliser un fichier XML ou tout programmer en Java.

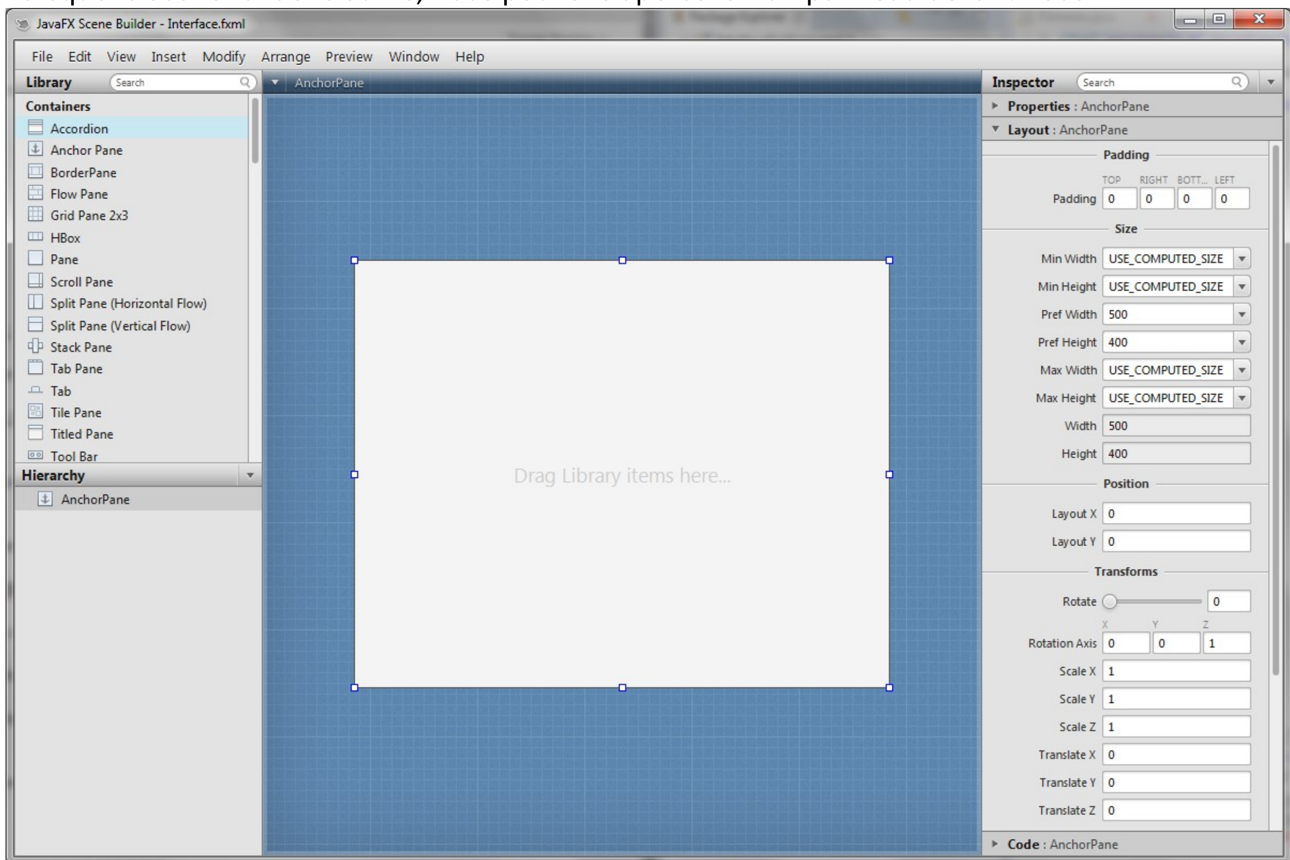
Notre SceneBuilder étant notre éditeur de fichier XML, nous n'aurons pas à toucher à proprement

dit à se langage.

Pour créer un fichier XML, clique droit sur notre *package "affichage" | New | Other | New FXML Document.*

Il ne reste plus qu'à l'ouvrir avec le SceneBuilder, pour se faire clique droit sur le fichier et "Open with SceneBuilder".

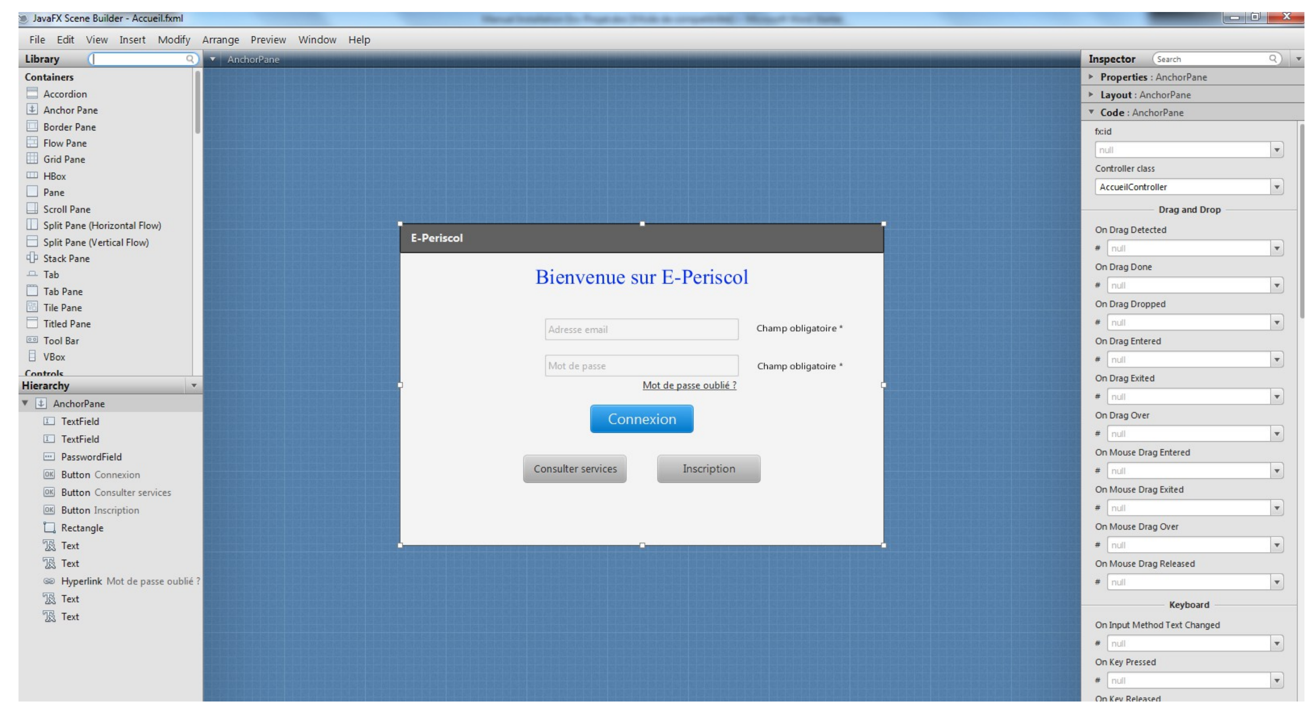
Lorsque le SceneBuilder s'ouvre, nous pouvons apercevoir un panneau devant nous :



Il ne nous reste plus qu'à créer l'interface de notre choix en ajoutant les éléments qui sont sur la gauche, le SceneBuilder est très intuitif, il suffit de quelques minutes pour comprendre son fonctionnement.

Après avoir fait notre interface, on peut commencer à coder nos classes java pour la rendre interactive.

L'interface est contrôlée par une classe java créée dans le projet :



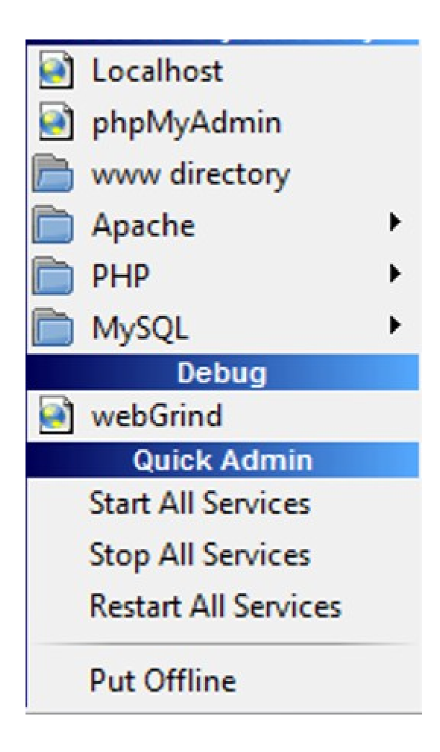
Il est donc normal que notre interface soit contrôlée par la classe AccueilController, puisque c'est elle qui communique avec le serveur JBoss.

III. Manuel d’installation de la base de données – Baptiste Cahuziere

1) Présentation:

La base de données du projet Incicom2 est une base de données MySQL, Réalisée grâce au logiciel Wampserver. En plus d'intégrer une base de données MySQL, elle intègre aussi un serveur « Apache » et une application WEB de gestion de base de données MySQL « phpMyAdmin ».

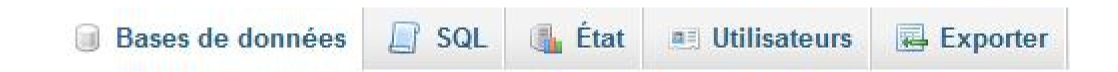
2) Installation de WAMP et importation de la base de données.



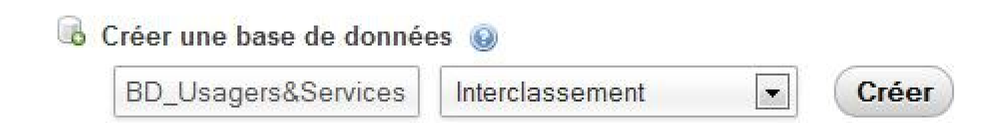
Il n’est normalement pas nécessaire de mettre Wampserver car il est déjà installé sur les PC fixe. Mais s’il n'est pas installé, il faut le télécharger sur internet et l'installer (ici la version 2.1).

Maintenant l’importation de la base de données. Pour cela il faut lancer phpMyAdmin en faisant un clic gauche sur l'icône de WAMP dans le menu des icônes systèmes.

Pour commencer, avant de remplir les informations concernant la base de données ou encore importer un fichier SQL il faut créer une base de données comme ci-dessous.



Bases de données



Une fois la base de données créée, nous pouvons insérer les différentes tables qui la compose, comme ci-dessous :



Structure | SQL | Rechercher | Requête | Exporter | Importer | Opérations | Privilèges | Procédures stockées

Table	Action	Lignes	Type	Interclassement	Taille	Perte
activités_culturelles	Afficher Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8_general_ci	1 Kio	-
enfants	Afficher Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8_general_ci	1 Kio	-
sports	Afficher Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8_general_ci	1 Kio	-
3 tables	Somme	0	MyISAM	utf8_general_ci	3 Kio	0 o

Dans chaque tables, nous entrons les attributs souhaités :

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	ID_Enfant	int(11)			Non	*Aucune*		Modifier Supprimer Affiche les valeurs distinctes Primaire plus
2	nom	varchar(50)	utf8_general_ci		Non	*Aucune*		Modifier Supprimer Affiche les valeurs distinctes Primaire plus
3	prenom	varchar(50)	utf8_general_ci		Non	*Aucune*		Modifier Supprimer Affiche les valeurs distinctes Primaire plus
4	age	int(11)			Non	*Aucune*		Modifier Supprimer Affiche les valeurs distinctes Primaire plus
5	sexe	varchar(10)	utf8_general_ci		Non	*Aucune*		Modifier Supprimer Affiche les valeurs distinctes Primaire plus
6	DateNaissance	datetime			Non	*Aucune*		Modifier Supprimer Affiche les valeurs distinctes Primaire plus

Fichier à importer :

Le fichier peut être comprimé (gzip, zip) ou non.
Le nom du fichier comprimé doit se terminer par **.[format].[compression]**. Exemple: **.sql.zip**

Parcourir : Parcourir... Aucun fichier sélectionné. (Taille maximum: 2 048Kio)

Jeu de caractères du fichier : utf-8

Importation partielle :

☑ Permettre l'interruption de l'importation si la limite de temps configurée dans PHP est sur le point d'être atteinte. *(Ceci pourrait aider à détriment du respect des transactions.)*

Nombre de lignes à ignorer à partir de la première ligne : 0

Format :

SQL

Options spécifiques au format :

Mode de compatibilité SQL : NONE

☑ Ne pas utiliser AUTO_INCREMENT pour la valeur zéro

Exécuter

Il est possible d'importer ou encore exporter, dans la base de données que nous venons de créer, des fichiers SQL édités auparavant. Voici un exemple ci-dessous :

III. Introduction au développement Android - Romain Gaillard:

Pré-requis :

Vérifier que vous avez installé le JDK (pour « Java Development Kit »). Si ce n'est pas le cas, rendez-vous sur http://www.oracle.com/technetwork/java/javase/downloads/index.html et cliquez sur **Download** à côté de **Java SE 6/7 Update xx.**

1) Installation du SDK Android :

Un SDK, est un kit de développement afin de vous permettre de développer des applications pour un environnement précis, ici Android.
Pour se le procurer, rendez-vous sur http://developer.android.com/sdk/index.html et cliquer sur *USE AN EXISTING IDE* puis sur *Download the SDK Tools.*

Au premier lancement du SDK, il faut installer les paquets dont vous aurez besoin.

Vous aurez besoin, des paquets *Tools* et *Extras*. Ensuite il faut sélectionner le paquet correspondant à la version d'Android pour laquelle vous souhaitez que votre application

soit compatible.

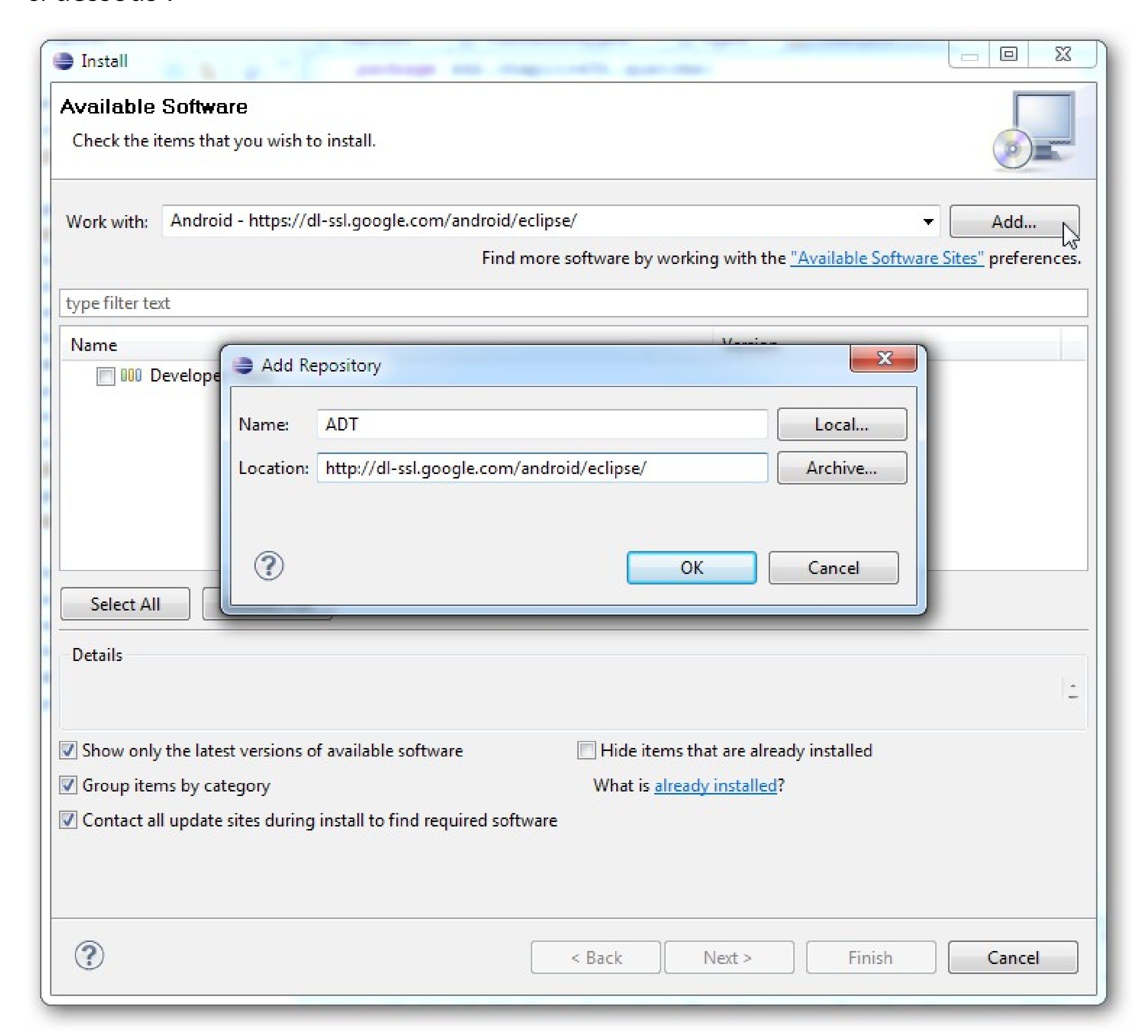
Une fois votre choix effectué, un écran vous demandera de confirmer que vous souhaitez bien télécharger ces éléments-là. Cliquez sur **Accept All** puis sur **Install** pour continuer.

2) Préparation de l'environnement Eclispe :

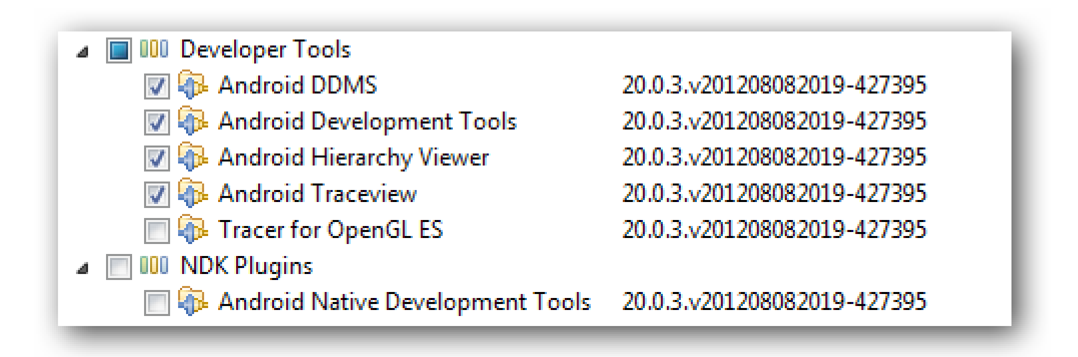
Rappel: Si vous ne possédez pas Eclipse, vous pouvez le télécharger à cette adresse : www.eclipse.org/downloads/

Pour développer une application Android avec Eclipse il est nécessaire de télécharger le plug-in *Android Development Tools* ***(ADT)***.Il permettra de créer les fichiers avec les bases, mais aussi tester, déboguer, et exporter vos projets au format **APK**. *(pour publier les applications)*

Dans Eclipse, allez dans **Help** puis dans **Install New Softwares**. A côté du champ **Work with:** cliquez sur le bouton **Add...** situé juste à côté. Dans l'encart **Name** écrivez *par exemple ADT* et, dans **location**, mettez cette adresse https://dl-ssl.google.com/android/eclipse/ comme-ci dessous :

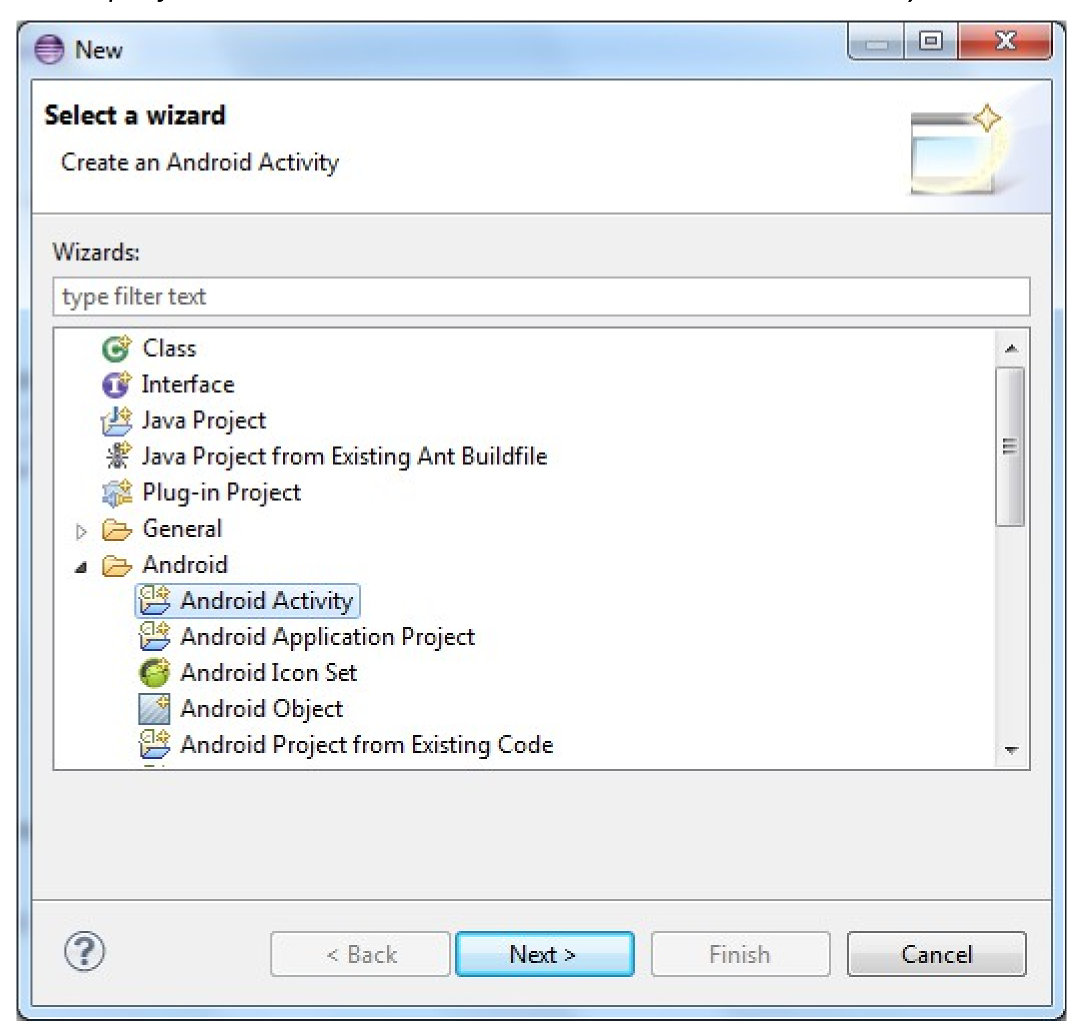


Ensuite télécharger les modules suivants :

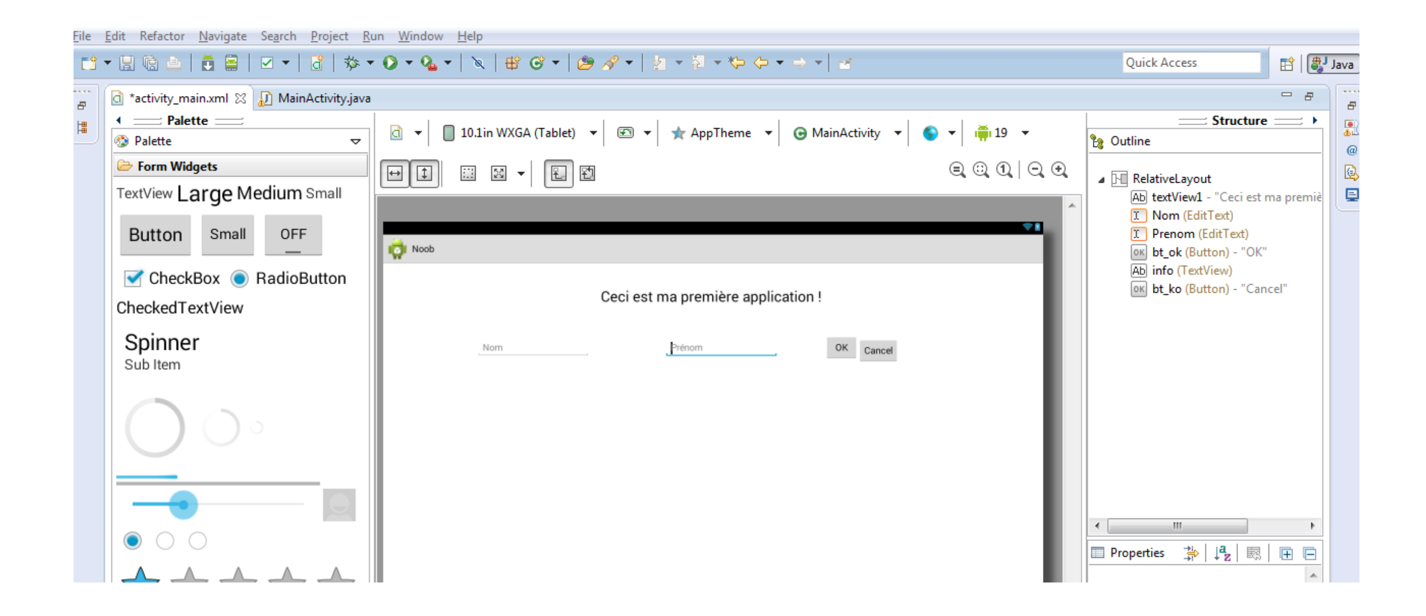


3) Création d'un projet :

Dans Eclipse faîtes Files > New > Other... et sélectionner Android Activity.



Ensuite il faut choisir la version Android pour laquelle on veut créer l'application.
On obtient ensuite un fichier .xml dont le code peut-être généré automatiquement grâce à l'assistant d'Eclipse, comme ci-dessous :
Ainsi qu'un fichier .java pour gérer les événements.



4) Configuration d'un terminal :

Aperçu en direct : (sans installation)

Important :

Sous Windows, il est nécessaire d'installer les **drivers** de votre périphériques.
Pour cela, il faut aller dans le gestionnaire des périphériques, faîtes un clic-droit sur votre composant, et cliquer sur mettre à jour le pilote. Vous pouvez sélectionner le choix automatique où indiquer l'emplacement des pilotes à installer. En règle générale, ces pilotes ont été téléchargés sur votre machine via le SDK.
Voici leur emplacement: En partant du répertoire où vous avez installé le SDK : *\android-sdk\extras\google\usb_driver.*

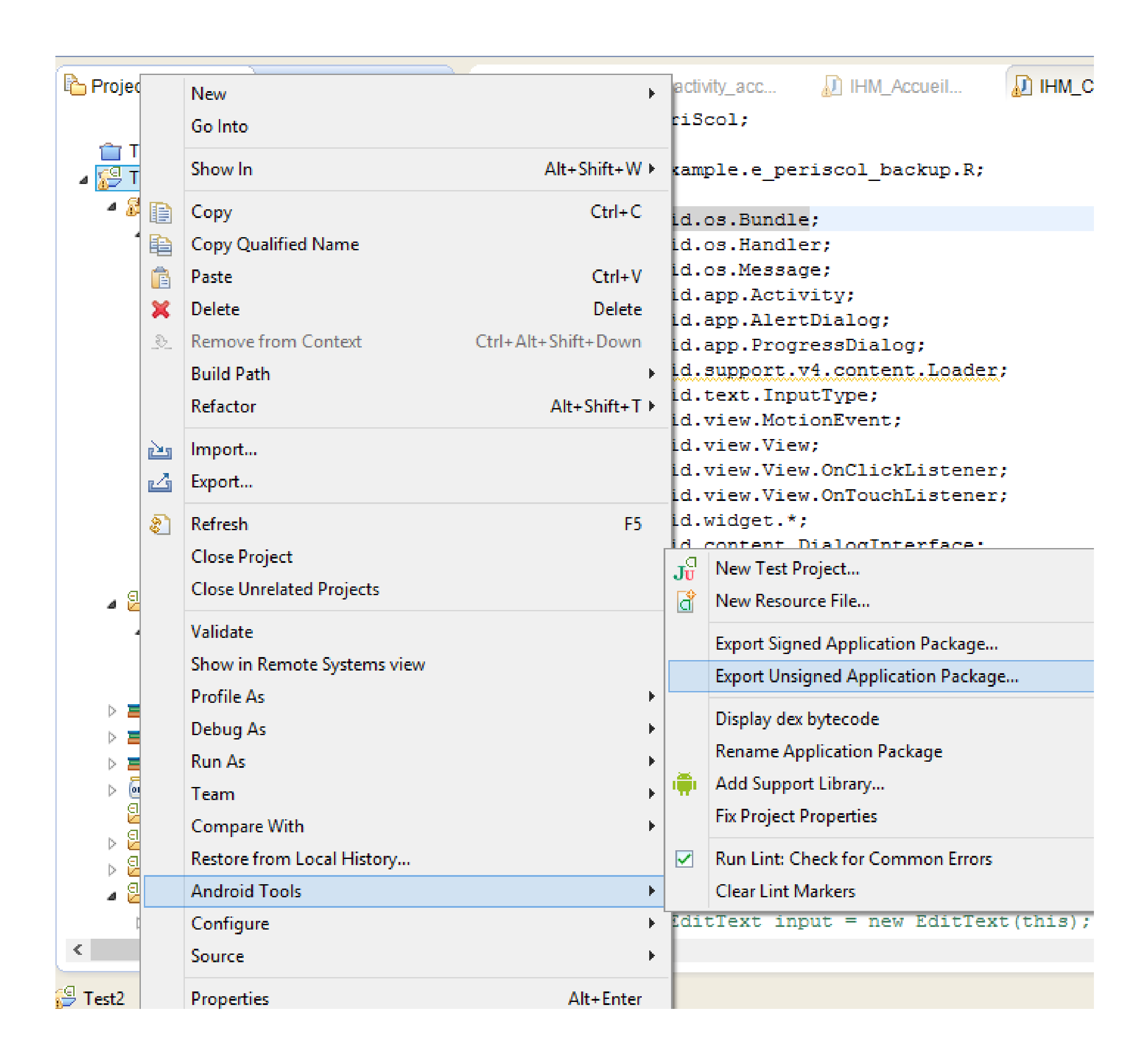
Ensuite, pour tester votre application au fur et à mesure du développement sous Eclipse, il faut connecter votre périphérique en USB et activer le mode Débogage USB *(Paramètres > Applications > Développement ou Paramètres >* Options pour les développement*)*



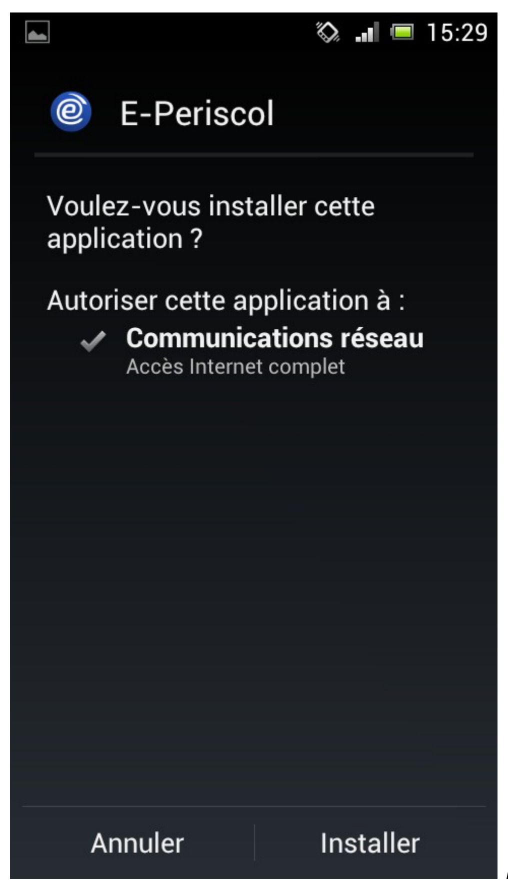
Remarque :
Il est également possible d'installer un émulateur de téléphone. Pour plus de renseignement: *http://fr.openclassrooms.com/informatique/cours/creez-des-applications-pour-android/installation-et-configuration-des-outils*

Application exécutable : (avec installation)
Tout d'abord il est nécessaire d'indiquer à votre périphérique d'accepter les applications de source inconnu. *(Autre que provenant du Play Store).*
Pour ce faire, aller dans Paramètres > Sécurité > Sources Inconnues ou Paramètres > Applications > Sources Inconnues. (suivant les versions d'Android).

Pour avoir un fichier exécutable, il faut faire cette manipulation sous Eclipse:



On va exporter l'application en mode « non signé », et le fichier .apk est créé. *(Pour le mode signé, destiné à la publication sur le play store, il faut remplir un formulaire.)* Maintenant l'application peut être installée sur votre téléphone ou votre tablette Android. Pour cela, il faut mettre le fichier .apk dans un répertoire de votre mobile. Ensuite utiliser une application gestionnaire de fichier qui vous permet d'accéder au dossier dans lequel vous avez mis votre fichier .apk et cliquer dessus. L'installation commence.

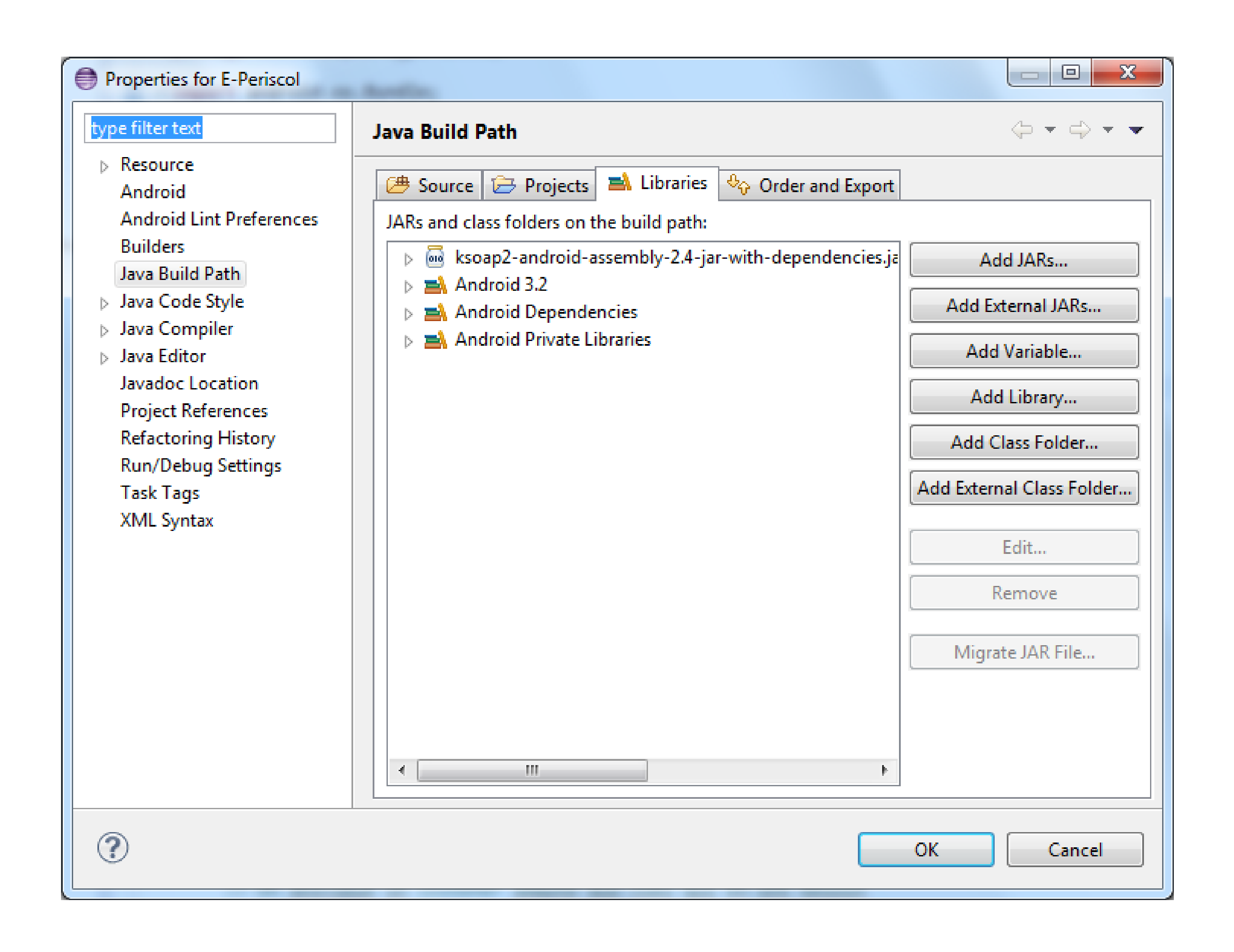


L'application demande les droits d'accès internet.

5) Ajout de la librairie KSOAP2 :

Il faut télécharger la librairie Ksoap2 android assembly (.jar) with depencies sur le site, pour le développement de la partie SOAP de notre application: http://code.google.com/p/ksoap2-android/downloads/detail?name=ksoap2-android-assembly-2.4-jar-with-dependencies.jar&

Ensuite on ajoute le .jar au projet :
Faîtes un clic-droit sur le dossier du projet > *properties* > *java build path* > *librairies*, ajouter le fichier jar.
(bouton *Add External JARs...*)


Properties for E-Periscol
type filter text
Resource
Android
Android Lint Preferences
Builders
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Project References
Refactoring History
Run/Debug Settings
Task Tags
XML Syntax
Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
ksoap2-android-assembly-2.4-jar-with-dependencies.ja
Android 3.2
Android Dependencies
Android Private Libraries
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
OK
Cancel

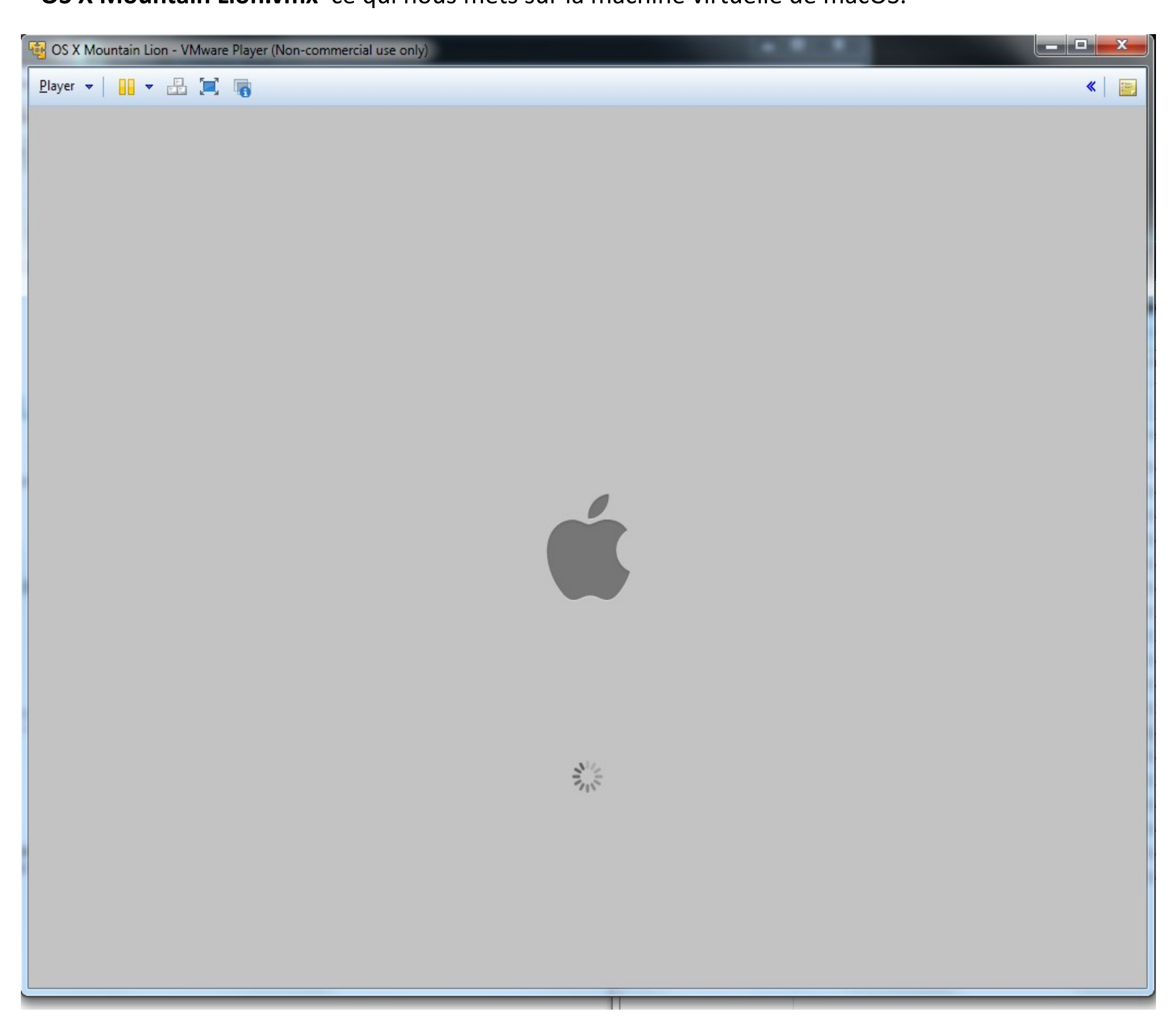
IV. Introduction au développement iOS – Cédric Cognard :

Pré-requis :

Pour créer une application **iOS** il vous faut travailler sur un système **macOS**. En effet, il faut utiliser **Xcode** qui est un environnement de développement logiciel et qui est uniquement disponible sur ce système d'exploitation.

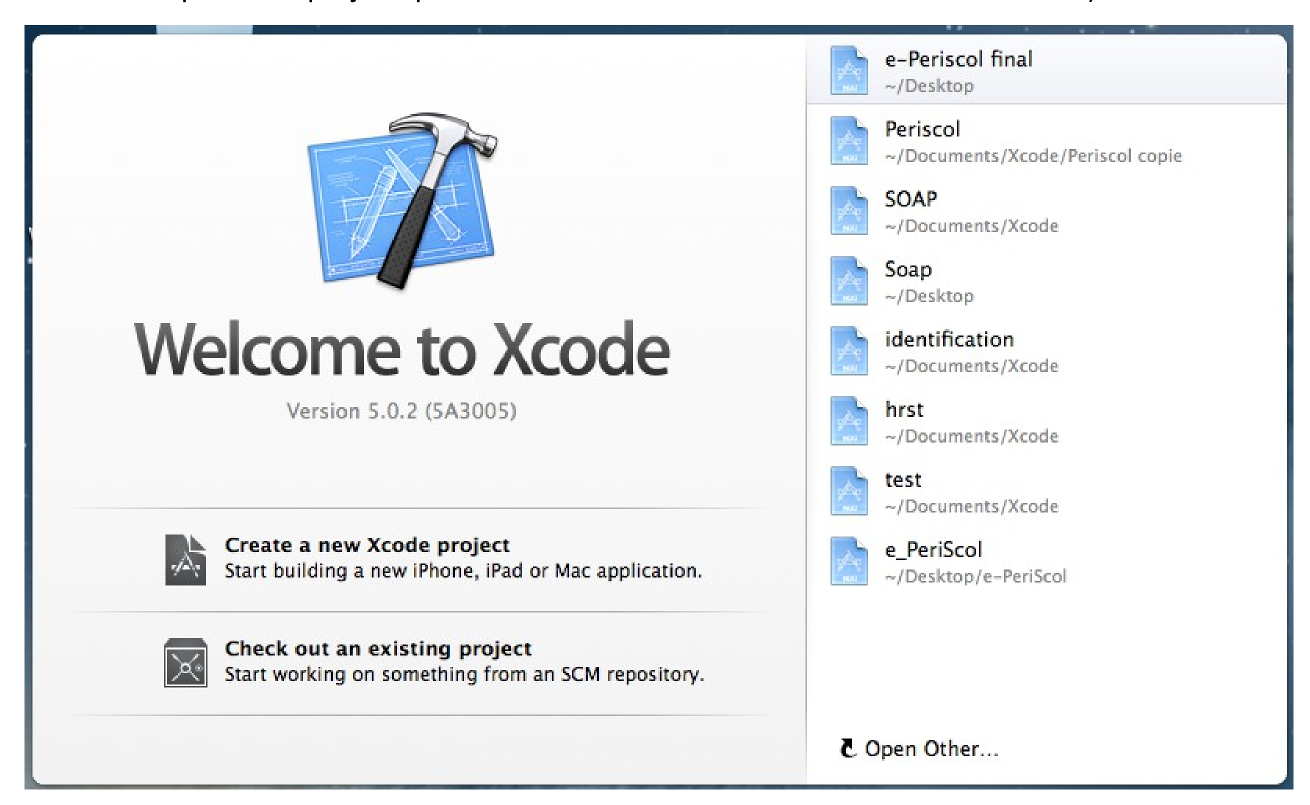
1) Machine virtuelle VMWare :

N'ayant pas à disponibilité de machine mac, nous utilisons une machine virtuelle qui a été préalablement installée sur le PC et qui utilise le logiciel **VMWare Player.** Il faut lancer
OS X Mountain Lion.vmx ce qui nous mets sur la machine virtuelle de macOS.

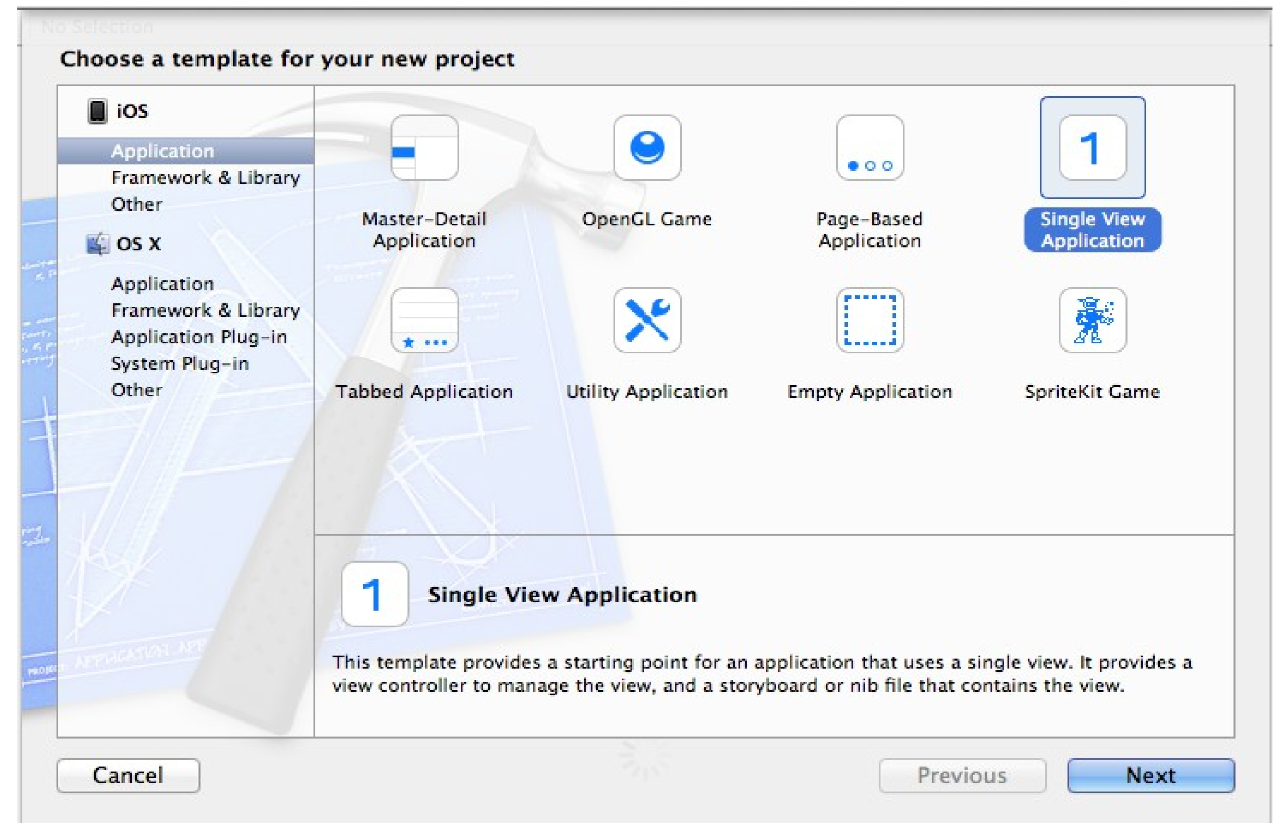


2) Création d'un projet :

Sur l'écran d'accueil de Xcode vous cliquez sur **Create a new Xcode project** (vous pouvez aussi choisir parmi les projets qui ont été récemment ouverts ou en rechercher un.)



Lors de la création d'un nouveau projet vous aurez le choix entre divers sorte de projets. On choisira ici un **Single View Application**.



Vous pouvez ensuite donner un nom à votre projet, ainsi que décider de son emplacement.

3) Configuration du projet :

Une fois votre projet ouvert vous avez la possibilité de changer certains paramètres comme la version **d'iOS** que vous souhaitez utiliser ainsi que si vous développez pour **iPhone** ou **iPad**. Pour cela, il faut cliquer sur le nom du projet dans la barre de gauche qui vous donne la hiérarchie de votre projet. Cette action vous affiche la page de configurations générales ainsi les paramètres du **Build**.

e-Periscol final
2 targets, iOS SDK 7.0
e-Periscol final
AppDelegate.h
AppDelegate.m
InsViewController.h
InsViewController.m
EnfantViewController.h
EnfantViewController.m
BienvViewController.h
Main_iPad.storyboard
Main_iPhone.storyboard
BienvViewController.m
Accesseur_WS.h
Accesseur_WS.m
Images.xcassets
TBXML
TBXML.h
TBXML.m
WSDL
Actions_ClientService.h
Actions_ClientService.m
NSDate+IS...01Parsing.h
NSDate+IS...01Parsing.m
NSDate+IS...Unparsing.h
NSDate+IS...nparsing.m
USAdditions.h
USAdditions.m
USGlobals.h
USGlobals.m
xs.h
xs.m
Supporting Files
e-Periscol finalTests
Frameworks
Products

e-Periscol final
e-Periscol final | General | Capabilities | Info | Build Settings | Build Phases | Build Rules

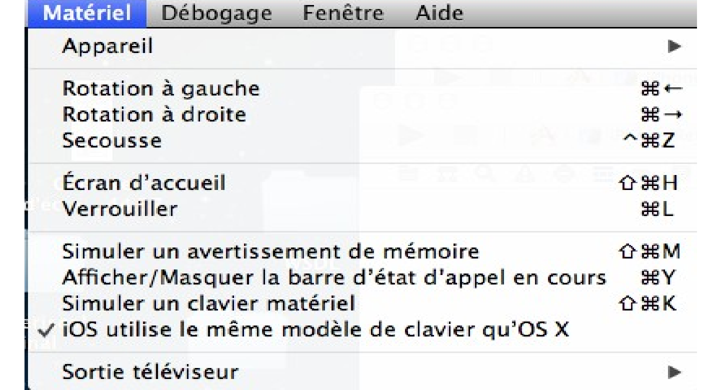
Identity
Bundle Identifier: irist.e-Periscol-final
Version: 1.0
Build: 1.0
Team: None
No matching code signing identity found
No codesigning identities (i.e. certificate and private key pairs) matching "iPhone Developer" were found. Xcode can resolve this issue by downloading a new provisioning profile from the Member Center.
Fix Issue

Deployment Info
Deployment Target: 7.0
Devices: Universal
iPhone | iPad
Main Interface: Main_iPhone
Device Orientation: Portrait, Upside Down, Landscape Left, Landscape Right
Status Bar Style: Default
Hide during application launch

Ici nous sommes sur un projet nommé **e-Periscol final**, qui est une application **iPhone** version 7.0

4) Utilisation du simulateur :

L'environnement **Xcode** est doté d'un **simulateur** qui vous permet de tester virtuellement vos applications.



Il permet de simuler toute les interactions possible avec **iPhone** et ainsi vérifier que vos fonctionnalités marche.

L'interface et l'utilisation sont les mêmes que sur un véritable **iPhone**. On peut configurer quel type d'**iPhone** est simuler, ici nous avons un **iPhone Retina** de 3.5 pouces qui fonctionne avec l'**iOS** 7.0.3.

